This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

CLAIMS

1. (Currently Amended) A method for verifying bus performance in a multiple initiator environment, a first initiator implementing the method, comprising:

generating a key data pattern including a key header and a pattern, the key header including data identifying the first initiator;

writing the key data pattern to an echo buffer of a target;

reading the key data pattern; and

examining the key header to ascertain a level of communication integrity of a physical connection with the target, the examining determining a throughput capability of the physical connection.

2. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 1, wherein generating the key header includes:

generating a byte 0;

generating a byte 1;

generating a byte 2; and

generating a byte 3.

3. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 2, wherein the byte 0 is an ID byte, the byte 1 is a host ID, the byte 2 is a logical negation of the host ID, and byte 3 is a logical negation of the ID byte.

4. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 3, wherein the ID byte is a manufacturer signature ID, and the host ID is an initiator ID.

5. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 1, wherein examining the key header includes one of:

determining whether the echo buffer returns an error indication;

determining whether data of the key header has been changed; or

determining whether the data in the key header specifically indicates a collision with data from another initiator using a same key header system.

6. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 5, wherein the determining of whether data of the key header has been changed occurs when the multiple initiators are heterogeneous.

7. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 5, wherein the determining of whether the data in the key header specifically indicates the collision occurs when the multiple initiators are homogeneous.

8. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 5, wherein when it is determined that the error indication is returned from the echo buffer, the first initiator being configured to rewrite the key data pattern to the echo buffer, the rewriting being performed for a set number of times before an adjustment is made to the level of communication integrity of the physical connection with the target.

9. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 6, wherein when it is determined that the data of the key header has been changed, the first initiator being configured to rewrite the key data pattern to the echo buffer, the rewriting being performed for a set number of times before an adjustment is made to the level of communication integrity of the physical connection with the target.

10. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 7, wherein when it is determined that the data in the key header specifically indicates the collision with data from another initiator using the same key header system, the first initiator being configured to rewrite the key data pattern to the echo buffer, the rewriting being performed for a set number of

times before an adjustment is made to the level of communication integrity of the physical connection with the target.

11. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 7, wherein the collision occurs when a byte 0 matches a specific manufacturer ID, a byte 1 does not match the first initiator's ID, a byte 2 is a logical negation of byte 1, and a byte 3 is a logical negation of byte 0.

12. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 6, wherein when it is determined that data of the key header has been changed, it is assumed that a collision occurred.

13. (Original) A method for verifying bus performance in a multiple initiator environment as recited in claim 1, wherein writing the key data pattern includes:

sending linked commands to the echo buffer to prevent the echo buffer from receiving data from another initiator, the linked commands being configured to link write and read commands and to disable a SCSI disconnection.

14. (Currently Amended) A computer implemented method for verifying bus performance in a multiple initiator environment that includes at least a first initiator and a second initiator in communication with a target device, the method comprising:

generating a key data pattern including a key header, the key header including data identifying the first initiator;

sending a write echo buffer (WEB) command to write the key data pattern to an echo buffer of the target;

sending a read echo buffer (REB) command to the echo buffer, the REB command being configured to request a transmission of the key data pattern from the echo buffer to the first initiator; and

examining the key data pattern received from the echo buffer to ascertain a level of communication integrity of a physical connection between the first initiator and the target device, the examining determining a throughput capability of the physical connection.

15. (Original) A computer implemented method for verifying bus performance in a multiple initiator environment as recited in claim 14, wherein before the key data pattern is generated, the method includes:

sending an asynchronous inquiry to the target device, the asynchronous inquiry being configured to request a transmission of a valid data pattern from the target device and receiving the valid data pattern from the target device in response to the asynchronous inquiry; and

sending a synchronous inquiry to the target device, the synchronous inquiry being configured to request a faster transmission of another valid data pattern in order to negotiate an optimal throughput speed with the target device and receiving the another valid data pattern from the target device in response to the synchronous inquiry.

16. (Original) A computer implemented method for verifying bus performance in a multiple initiator environment as recited in claim 15, wherein after the sending of the synchronous inquiry, the method includes:

sending a read echo buffer description (REBD) command to the echo buffer of the target, the REBD command being configured to request information regarding a size of the echo buffer and whether the echo buffer supports collision detection.

17. (Original) A computer implemented method for verifying bus performance in a multiple initiator environment as recited in claim 14, further comprising:

detecting a data collision during the examining of the key data pattern received from the echo buffer; and

if a collision is detected, the method includes,

re-sending a WEB command with the key data pattern to the echo buffer, the re-sending being performed for a set number of times before an adjustment is made to the level of communication integrity of the physical connection between the first initiator and the target.

18. (Original) A computer implemented method for verifying bus performance in a multiple initiator environment as recited in claim 14, wherein generating the key header includes:

generating a byte 0;

generating a byte 1;

generating a byte 2;

generating a byte 3; and

generating a pattern.

19. (Original) A computer implemented method for verifying bus performance in a multiple initiator environment as recited in claim 18, wherein the byte 0 is an ID byte, the byte 1 is a host ID, the byte 2 is a logical negation of the host ID, and byte 3 is a logical negation of the ID byte.

20. (Original) A computer implemented method for verifying bus performance in a multiple initiator environment as recited in claim 19, wherein the ID byte is a manufacturer signature ID, and the host ID is an initiator ID.

21. (Original) A computer readable media having program instructions for verifying bus performance in a multiple initiator environment that includes at least a first initiator and a second initiator in communication with a target device, the computer readable media comprising:

program instructions for generating a key data pattern;

program instructions for sending a write echo buffer (WEB) command to write the key data pattern to an echo buffer of the target;

program instructions for sending a read echo buffer (REB) command to the echo buffer, the REB command being configured to request a transmission of the key data pattern from the echo buffer to the first initiator; and

program instructions for examining the key data pattern received from the echo buffer to ascertain a level of communication integrity of a physical connection between the first initiator and the target device.

22. (Original) A computer readable media as recited in claim 21, further comprising:

program instructions for detecting a data collision during the examining of the key data pattern received from the echo buffer; and

if a collision is detected, the method includes,

program instructions for re-sending a WEB command with the key data pattern to the echo buffer, the re-sending being performed for a set number of times before an adjustment is made to the level of communication integrity of the physical connection between the first initiator and the target.

23. (Original) A computer readable media as recited in claim 21, wherein program instructions for generating the key header includes:

program instructions for generating a byte 0;

program instructions for generating a byte 1;

program instructions for generating a byte 2;

program instructions for generating a byte 3; and

program instructions for generating a pattern.

24. (Original) A computer readable media as recited in claim 23, wherein the byte 0 is an ID byte, the byte 1 is a host ID, the byte 2 is a logical negation of the host ID, and byte 3 is a logical negation of the ID byte.